1Z0-809 Java SE 8 Programmer II Exam Questions and Answers PDF

Oracle 1Z0-809 Exam

DBExam.com

Get complete detail on 1Z0-809 exam guide to crack Java SE 8 Programmer II. You can collect all information on 1Z0-809 tutorial, practice test, books, study material, exam questions, and syllabus. Firm your knowledge on Java SE 8 Programmer II and get ready to crack 1Z0-809 certification. Explore all information on 1Z0-809 exam with number of questions, passing percentage and time duration to complete test.

How to Prepare for 1Z0-809 Exam on Java SE 8 Programmer II Certification?

Java SE 8 Programmer II (1Z0-809) preparation guide helps you to get focused on the exam. This guide also helps you to be on the 1Z0-809 exam track to get certified with a good score in the final exam.

1Z0-809 Java SE 8 Programmer II Exam Summary

Exam Name	Java SE 8 Programmer II
Exam Code	1Z0-809
Exam Product Version	Java SE 8

Exam Price	USD $245 (Pricing may vary by country or by localized currency)
Duration	120 minutes
Number of Questions	68
Passing Score	65%
Format	Multiple Choice Questions (MCQ)
Recommended Training	**Java SE 8 Programmer - Professional**
Schedule Exam	**Buy Oracle Training and Certification**
Sample Questions	**Oracle Certified Professional Java SE 8 Programmer (OCP)**
Recommended Practice	**1Z0-809 Online Practice Exam**

Exam Syllabus: 1Z0-809 Java SE 8 Programmer II

Java Class Design	- Implement encapsulation - Implement inheritance including visibility modifiers and composition - Implement polymorphism - Override hashCode, equals, and toString methods from Object class - Create and use singleton classes and immutable classes - Develop code that uses static keyword on initialize blocks, variables, methods, and classes

Advanced Java Class Design	- Develop code that uses abstract classes and methods - Develop code that uses the final keyword - Create inner classes including static inner class, local class, nested class, and anonymous inner class - Use enumerated types including methods, and constructors in an enum type - Develop code that declares, implements and/or extends interfaces and use the @Override annotation. - Create and use Lambda expressions
Generics and Collections	- Create and use a generic class - Create and use ArrayList, TreeSet, TreeMap, and ArrayDeque objects - Use java.util.Comparator and java.lang.Comparable interfaces - Collections Streams and Filters - Iterate using forEach methods of Streams and List - Describe Stream interface and Stream pipeline - Filter a collection by using lambda expressions - Use method references with Streams
Lambda Built-in Functional Interfaces	- Use the built-in interfaces included in the java.util.function package such as Predicate, Consumer, Function, and Supplier - Develop code that uses primitive versions of functional interfaces - Develop code that uses binary versions of functional interfaces - Develop code that uses the UnaryOperator interface
Java Stream API	- Develop code to extract data from an object using peek() and map() methods including primitive versions of the map() method - Search for data by using search methods of the Stream classes including findFirst, findAny, anyMatch, allMatch, noneMatch - Develop code that uses the Optional class - Develop code that uses Stream data methods and calculation methods - Sort a collection using Stream API - Save results to a collection using the collect method and group/partition data using the Collectors class - Use flatMap() methods in the Stream API
Exceptions and Assertions	- Use try-catch and throw statements - Use catch, multi-catch, and finally clauses - Use Autoclose resources with a try-with-resources statement - Create custom exceptions and Auto-closeable resources - Test invariants by using assertions

Use Java SE 8 Date/Time API	- Create and manage date-based and time-based events including a combination of date and time into a single object using LocalDate, LocalTime, LocalDateTime, Instant, Period, and Duration - Work with dates and times across timezones and manage changes resulting from daylight savings including Format date and times values - Define and create and manage date-based and time-based events using Instant, Period, Duration, and TemporalUnit
Java I/O Fundamentals	- Read and write data from the console - Use BufferedReader, BufferedWriter, File, FileReader, FileWriter, FileInputStream, FileOutputStream, ObjectOutputStream, ObjectInputStream, and PrintWriter in the java.io package.
Java File I/O (NIO.2)	- Use Path interface to operate on file and directory paths - Use Files class to check, read, delete, copy, move, manage metadata of a file or directory - Use Stream API with NIO.2
Java Concurrency	- Create worker threads using Runnable, Callable and use an ExecutorService to concurrently execute tasks - Identify potential threading problems among deadlock, starvation, livelock, and race conditions - Use synchronized keyword and java.util.concurrent.atomic package to control the order of thread execution - Use java.util.concurrent collections and classes including CyclicBarrier and CopyOnWriteArrayList - Use parallel Fork/Join Framework - Use parallel Streams including reduction, decomposition, merging processes, pipelines and performance.
Building Database Applications with JDBC	- Describe the interfaces that make up the core of the JDBC API including the Driver, Connection, Statement, and ResultSet interfaces and their relationship to provider implementations - Identify the components required to connect to a database using the DriverManager class including the JDBC URL - Submit queries and read results from the database including creating statements, returning result sets, iterating through the results, and properly closing result sets, statements, and connections
Localization	- Read and set the locale by using the Locale object - Create and read a Properties file - Build a resource bundle for each locale and load a resource bundle in an application
Assume the following:	1. Missing package and import statements: - If sample code do not include package or import

	statements, and the question does not explicitly refer to these missing statements, then assume that all sample code is in the same package, or import statements exist to support them. 2. No file or directory path names for classes: - If a question does not state the file names or directory locations of classes, then assume one of the following, whichever will enable the code to compile and run: - All classes are in one file - Each class is contained in a separate file, and all files are in one directory 3. Unintended line breaks: - Sample code might have unintended line breaks. If you see a line of code that looks like it has wrapped, and this creates a situation where the wrapping is significant (for example, a quoted String literal has wrapped), assume that the wrapping is an extension of the same line, and the line does not contain a hard carriage return that would cause a compilation failure. 4. Code fragments: - A code fragment is a small section of source code that is presented without its context. Assume that all necessary supporting code exists and that the supporting environment fully supports the correct compilation and execution of the code shown and its omitted environment. 5. Descriptive comments: - Take descriptive comments, such as "setter and getters go here," at face value. Assume that correct code exists, compiles, and runs successfully to create the described effect.

Oracle 1Z0-809 Certification Sample Questions and Answers

To make you familiar with Java SE 8 Programmer II (1Z0-809) certification exam structure, we have prepared this sample question set. We suggest you to try our **Sample Questions for Oracle OCA 1Z0-809 Certification** to test your understanding of Oracle 1Z0-809 process with the real Oracle certification exam environment.

1Z0-809 Java SE 8 Programmer II Sample Questions

01. What best describes a reduction?

a) A terminal operation where one element is returned from the prior step in a stream pipeline without reading all the elements

b) A terminal operation where a single value is generated by reading each element in the prior step in a stream pipeline
c) An intermediate operation where it mathematically divides each element in the stream
d) An intermediate operation where it filters the stream it receives

02. Which functional interface takes a double value and has a test() method?
a) Double Consumer
b) Double Predicate
c) Double Unary Operator
d) To Double Function

03. Which of the following statements about inheritance and object composition are correct?
a) Inheritance supports access to protected variables.
b) Object composition tends to promote greater code reuse than inheritance.
c) Inheritance relies on the has- a principle.
d) Object composition supports method overriding at runtime.
e) Object composition requires a class variable to be declared public or accessible from a public method to be used by a class in a different package.
f) Object composition is always preferred to inheritance.

04. How do you change the value of an instance variable in an immutable class?
a) Call the setter method.
b) Remove the final modifier and set the instance variable directly.
c) Use a method other than Option Call the setter method. or Remove the final modifier and set the instance variable directly..
d) You can’t.

05. When localizing an application, which type of data varies in presentation depending on locale?
a) Dates and Currencies
b) Currencies
c) Dates
d) Neither

06. Choose the class that is least likely to be marked Serializable.
a) A class that holds data about the amount of rain that has fallen in a given year
b) A class that manages the memory of running processes in an application
c) A class that stores information about apples in an orchard
d) A class that tracks the amount of candy in a gumball machine

07. How do you find out the locale of the running program?
a) Locale.get("default")
b) Locale.get(Locale.DEFAULT)

c) Locale.getDefault()
d) None of the above

08. What is a common reason for a stream pipeline not to run?
a) The source doesn't generate any items.
b) There are no intermediate operations.
c) The terminal operation is missing.
d) None of the above

09. Assuming / is the root directory, which of the following are true statements?
a) /home/parrot is an absolute path.
b) /home/parrot is a directory.
c) /home/parrot is a relative path.
d) The path pointed to from a File object must exist.
e) The parent of the path pointed to by a File object must exist.

10. Which of the following will print current time?
a) System.out.print(new LocalTime().now());
b) System.out.print(new LocalTime());
c) System.out.print(LocalTime.now());
d) System.out.print(LocalTime.today());
e) None of the above.

Solution:

QUESTION: 01 Answer: b	QUESTION: 02 Answer: b	QUESTION: 03 Answer: a, b, e	QUESTION: 04 Answer: d	QUESTION: 05 Answer: a
QUESTION: 06 Answer: b	QUESTION: 07 Answer: c	QUESTION: 08 Answer: c	QUESTION: 09 Answer: a	QUESTION: 10 Answer: c

How to Register for the 1Z0-809 Java SE 8 Programmer II Exam?

- **Purchase exam voucher** from Oracle University
- Register for an exam at **Buy Oracle Training and Certification**.